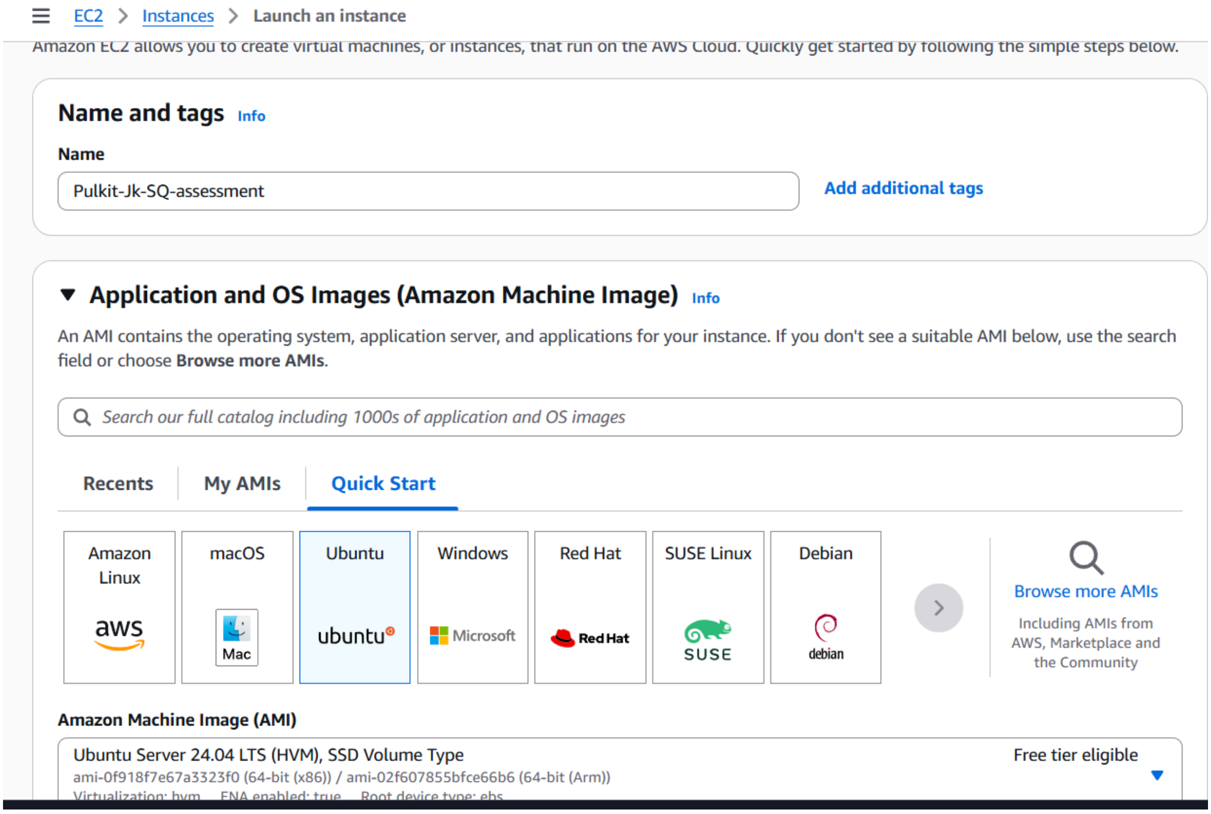
Assessment 2

Sonarcube integration using Jenkins

Part A

Step 1. Launching a EC2 Instance

a) Naming instance an selecting Ubuntu as AMI



EC2 > Instances > Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: Pulkit-Jk-SQ-assessment

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

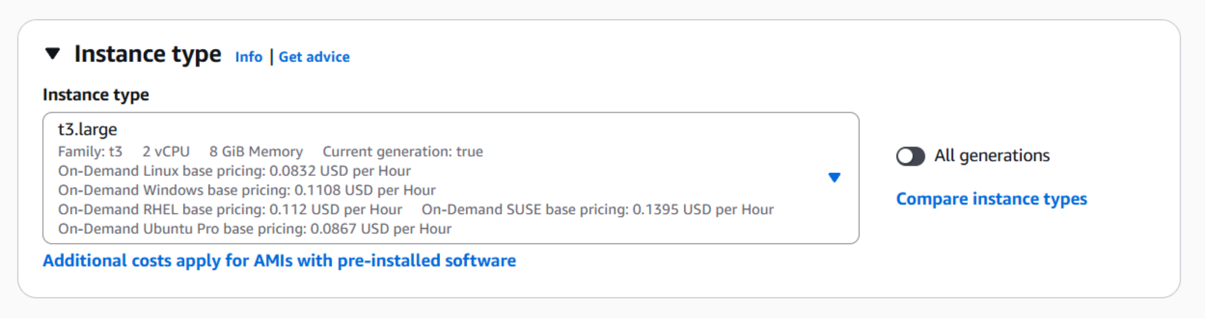
Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0f918f7e67a3323f0 (64-bit (x86)) / ami-02f607855bfce66b6 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

b) Selecting the instance type as t3.large

Note: It is prescribed to use t3.medium but is has less disk space thus the pipeline will not get deployed thus recommended to use t3.large and make Root storage to 30GiB



▼ Instance type Info | Get advice

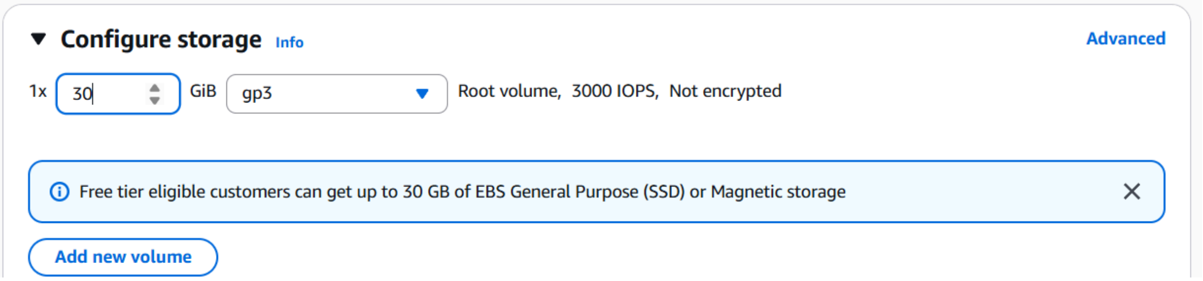
Instance type

t3.large
Family: t3 2 vCPU 8 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0832 USD per Hour
On-Demand Windows base pricing: 0.1108 USD per Hour
On-Demand RHEL base pricing: 0.112 USD per Hour On-Demand SUSE base pricing: 0.1395 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0867 USD per Hour

Additional costs apply for AMIs with pre-installed software

All generations

Compare instance types



▼ Configure storage Info

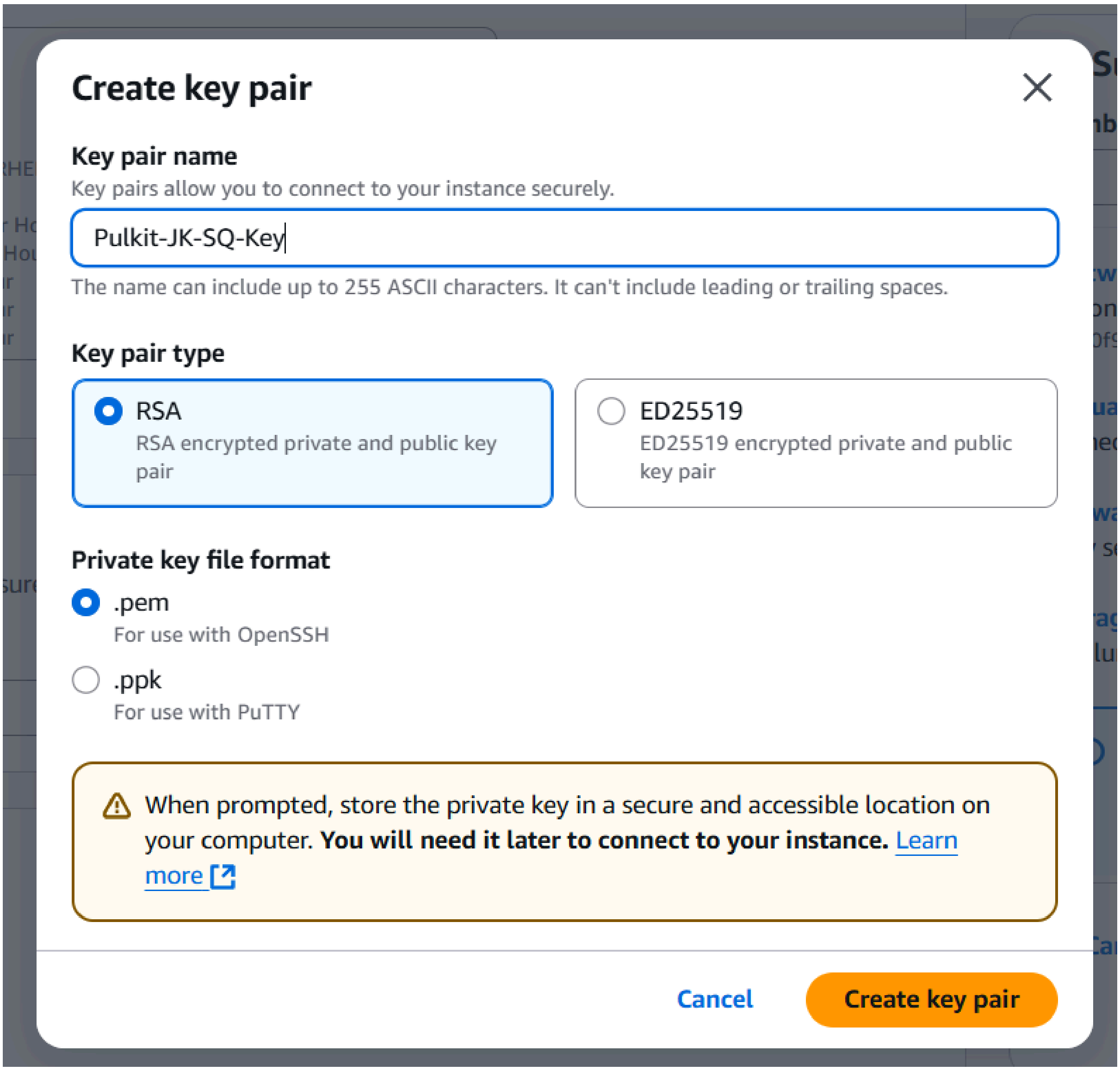
Advanced

1x 30 GiB gp3 Root volume, 3000 IOPS, Not encrypted

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✕

Add new volume

c) Creating a key pair with RSA as key pair type and .pem as key file format



Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

Pulkit-JK-SQ-Key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair

Private key file format

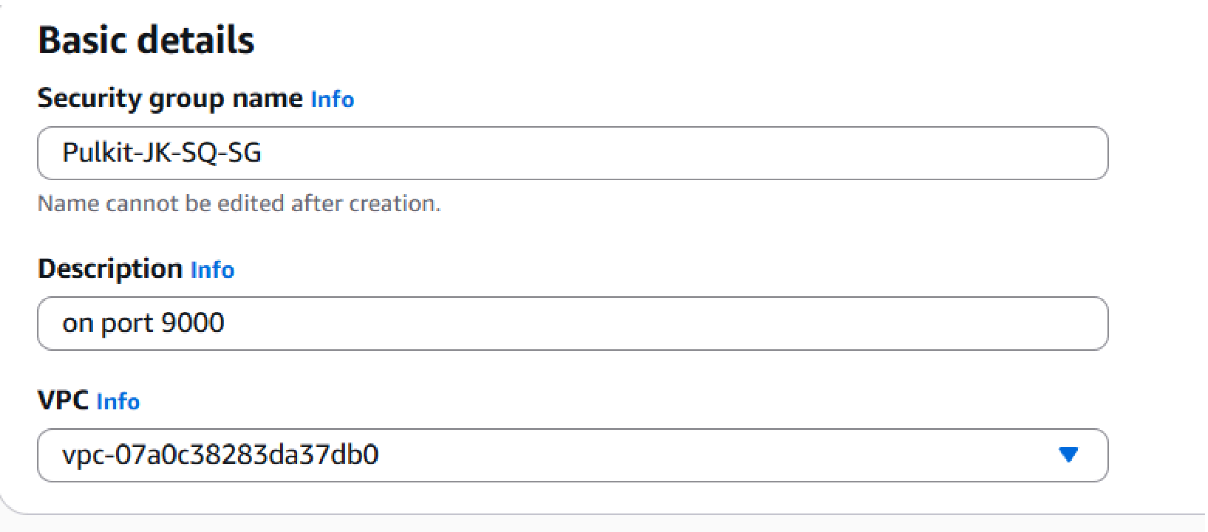
● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

d) Creating Security group

- **Naming it and selecting default available vpc**



Basic details

Security group name Info

Pulkit-JK-SQ-SG

Name cannot be edited after creation.

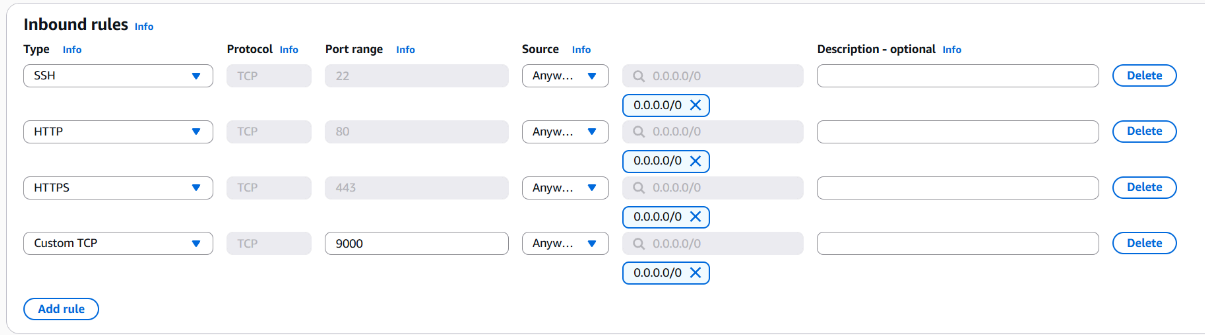
Description Info

on port 9000

VPC Info

vpc-07a0c38283da37db0 ▼

- **Creating Inbound rules with port 9000 allowed**

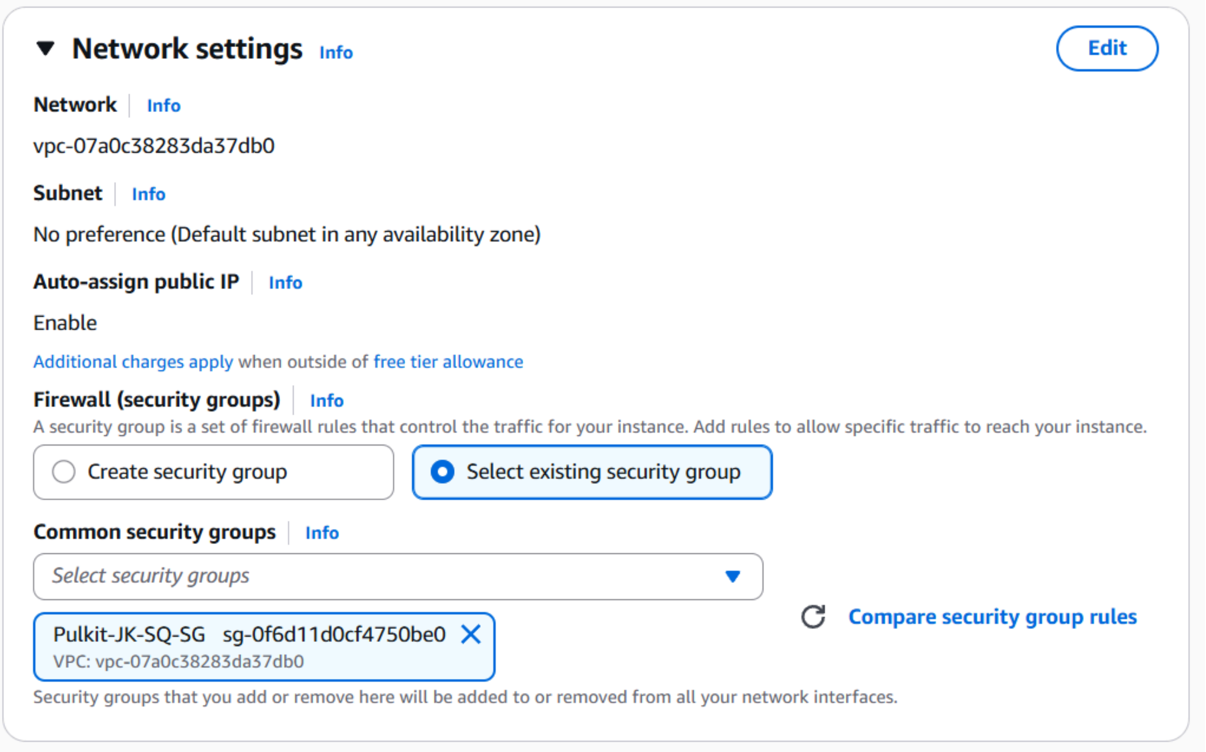


Inbound rules Info

Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
SSH ▼	TCP	22	Anyw... ▼	0.0.0.0/0 0.0.0.0/0 ✕		Delete
HTTP ▼	TCP	80	Anyw... ▼	0.0.0.0/0 0.0.0.0/0 ✕		Delete
HTTPS ▼	TCP	443	Anyw... ▼	0.0.0.0/0 0.0.0.0/0 ✕		Delete
Custom TCP ▼	TCP	9000	Anyw... ▼	0.0.0.0/0 0.0.0.0/0 ✕		Delete

Add rule

e) Selecting the security group that we just created



▼ **Network settings** Info

Network | Info

vpc-07a0c38283da37db0

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group ● Select existing security group

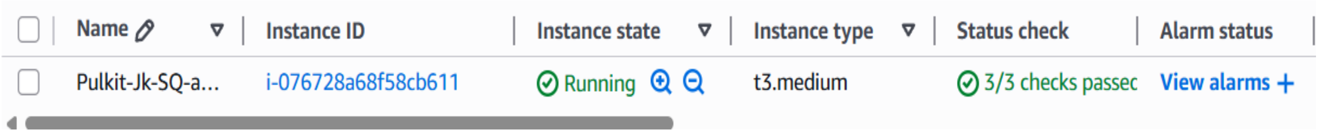
Common security groups | Info

Select security groups ▼

Pulkit-JK-SQ-SG sg-0f6d11d0cf4750be0 ✕
VPC: vpc-07a0c38283da37db0

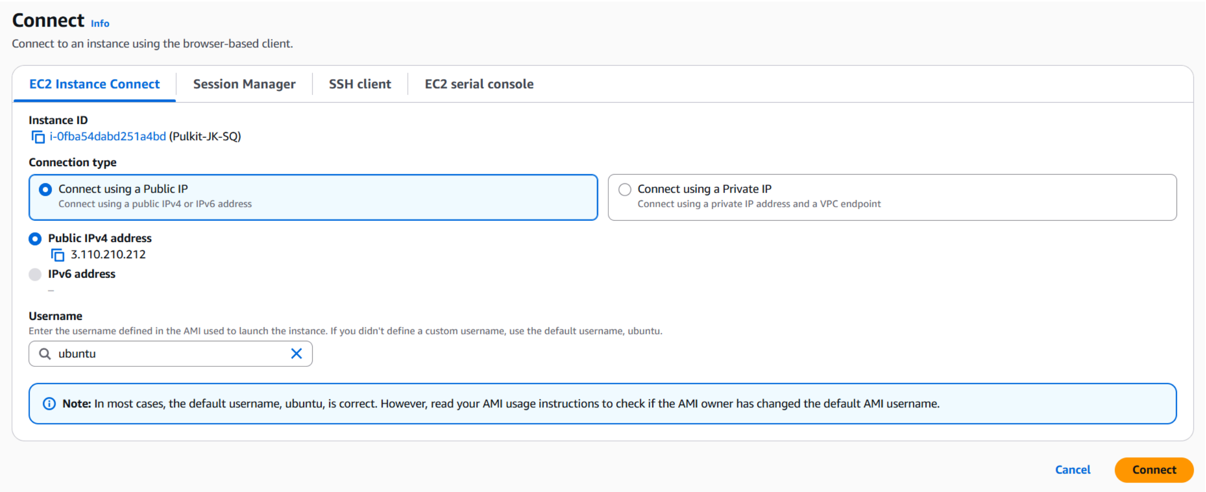
Security groups that you add or remove here will be added to or removed from all your network interfaces.

f) Launching EC2



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	Pulkit-Jk-SQ-a...	i-076728a68f58cb611	✓ Running	t3.medium	✓ 3/3 checks passec	View alarms +

Step 2. Connecting it to ssh



Connect Info

Connect to an instance using the browser-based client.

Instance ID
i-0fba54dabd251a4bd (Pulkit-JK-SQ)

Connection type

● Connect using a Public IP
Connect using a public IPv4 or IPv6 address

○ Connect using a Private IP
Connect using a private IP address and a VPC endpoint

● Public IPv4 address
3.110.210.212

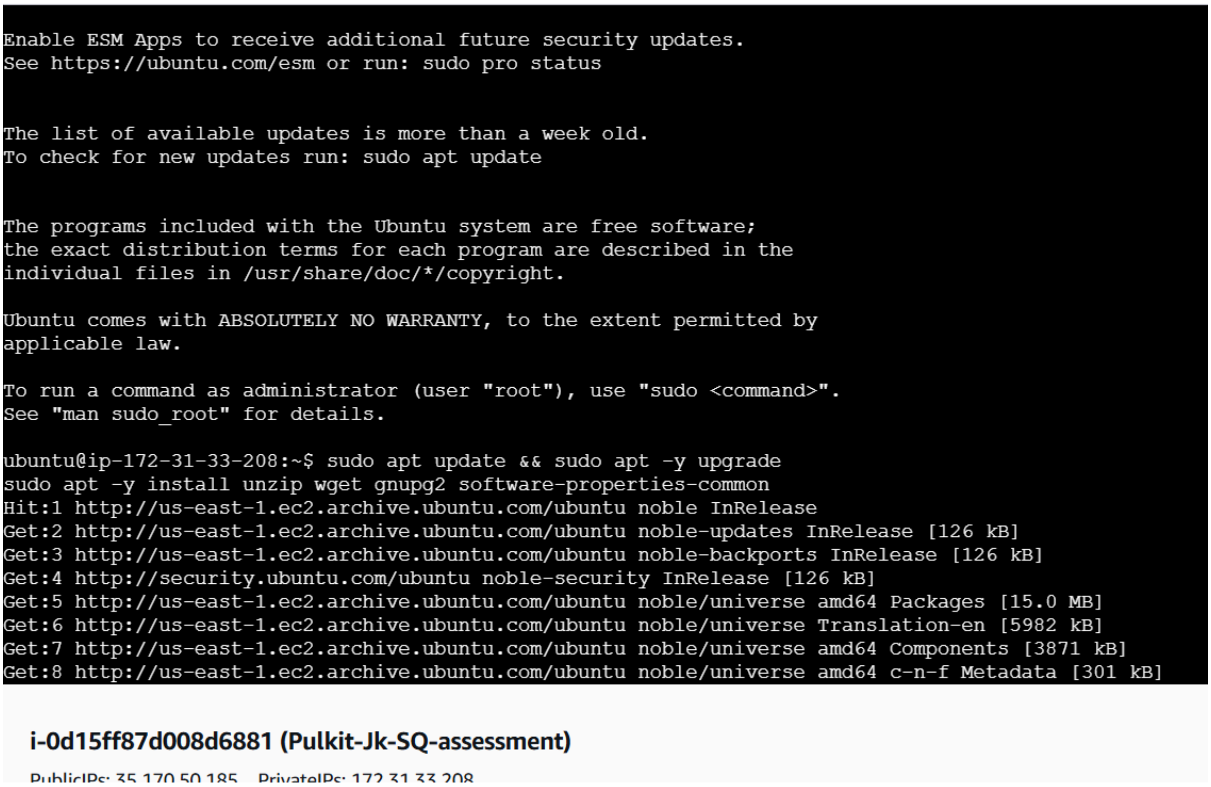
○ IPv6 address
–

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Step 3. Preparing ssh with basic updates

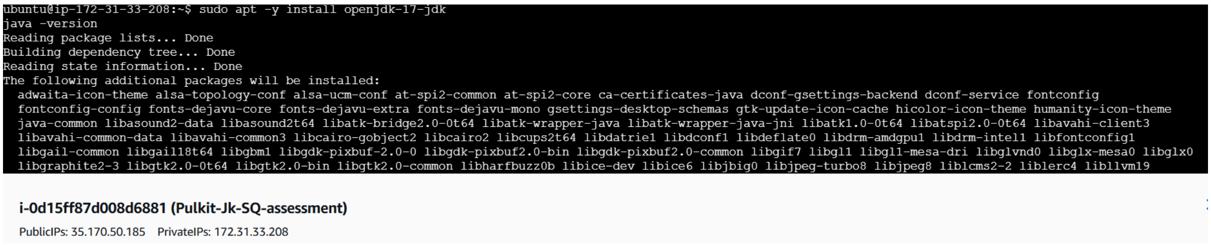


```
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-33-208:~$ sudo apt update && sudo apt -y upgrade
sudo apt -y install unzip wget gnupg2 software-properties-common
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
```

i-0d15ff87d008d6881 (Pulkit-Jk-SQ-assessment)

PublicIPs: 35.170.50.185 PrivateIPs: 172.31.33.208

Step 4. Installing Java jdk

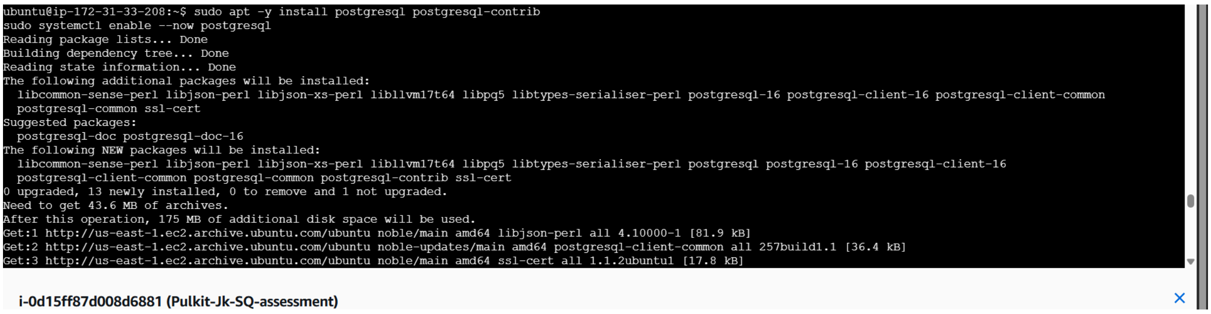


```
ubuntu@ip-172-31-33-208:~$ sudo apt -y install openjdk-17-jdk
java -version
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig
  fontconfig-config fonts-dejavu-core fonts-dejavu-extra fonts-dejavu-mono gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme
  java-common libasound2-data libasound2t64 libatk-bridge2.0-0t64 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64 libavahi-client3
  libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcups2t64 libdatrie1 libdconf1 libdeflate0 libdrm-amdgpu1 libdrm-intel1 libfontconfig1
  libgail-common libgail18t64 libgbm1 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-mesa-dri libglvnd0 libglx-mesa0 libglx0
  libgraphite2-3 libgtk2.0-0t64 libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice-dev libice6 libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 liblerc4 libllvm19
```

i-0d15ff87d008d6881 (Pulkit-Jk-SQ-assessment)

PublicIPs: 35.170.50.185 PrivateIPs: 172.31.33.208

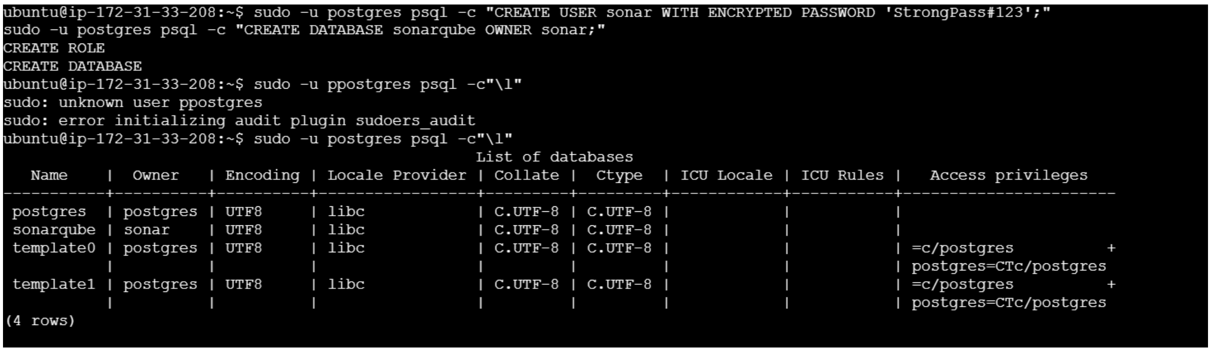
Step 5. Installing PostgreSQL



```
ubuntu@ip-172-31-33-208:~$ sudo apt -y install postgresql postgresql-contrib
sudo systemctl enable --now postgresql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm17t64 libpq5 libtypes-serialiser-perl postgresql-16 postgresql-client-16 postgresql-client-common
  postgresql-common ssl-cert
Suggested packages:
  postgresql-doc postgresql-doc-16
The following NEW packages will be installed:
  libcommon-sense-perl libjson-perl libjson-xs-perl libllvm17t64 libpq5 libtypes-serialiser-perl postgresql postgresql-16 postgresql-client-16
  postgresql-client-common postgresql-common postgresql-contrib ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 1 not upgraded.
Need to get 43.6 MB of archives.
After this operation, 175 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libjson-perl all 4.10000-1 [81.9 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 postgresql-client-common all 257build1.1 [36.4 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntu1 [17.8 kB]
```

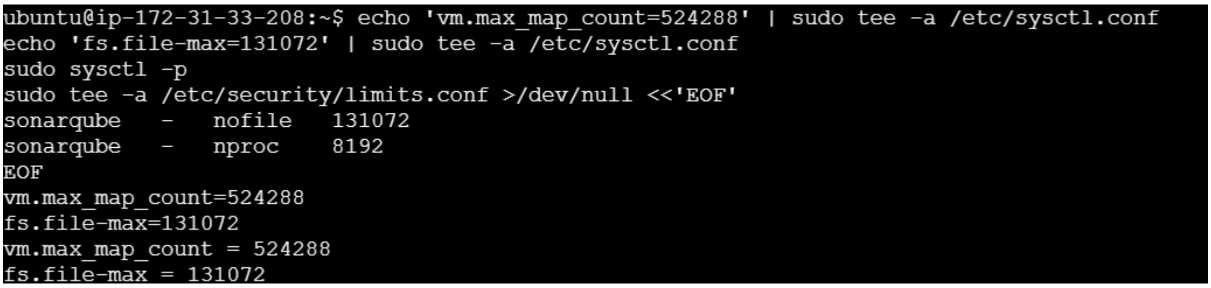
i-0d15ff87d008d6881 (Pulkit-Jk-SQ-assessment)

Step 6. Creating DataBase and user



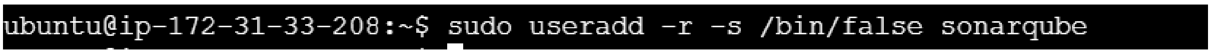
```
ubuntu@ip-172-31-33-208:~$ sudo -u postgres psql -c "CREATE USER sonar WITH ENCRYPTED PASSWORD 'StrongPass#123';"
sudo -u postgres psql -c "CREATE DATABASE sonarqube OWNER sonar;"
CREATE ROLE
CREATE DATABASE
ubuntu@ip-172-31-33-208:~$ sudo -u ppostgres psql -c"\l"
sudo: unknown user ppostgres
sudo: error initializing audit plugin sudoers_audit
ubuntu@ip-172-31-33-208:~$ sudo -u postgres psql -c"\l"
                                                 List of databases
   Name    |  Owner   | Encoding | Locale Provider | Collate |  Ctype  | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+---------+---------+------------+-----------+-----------------------
 postgres  | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |            |           |
 sonarqube | sonar    | UTF8     | libc            | C.UTF-8 | C.UTF-8 |            |           |
 template0 | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |         |         |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |            |           | =c/postgres          +
           |          |          |                 |         |         |            |           | postgres=CTc/postgres
(4 rows)
```

Step 7. Defining Linux limits and parameters



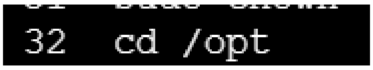
```
ubuntu@ip-172-31-33-208:~$ echo 'vm.max_map_count=524288' | sudo tee -a /etc/sysctl.conf
echo 'fs.file-max=131072' | sudo tee -a /etc/sysctl.conf
sudo sysctl -p
sudo tee -a /etc/security/limits.conf >/dev/null <<'EOF'
sonarqube   -   nofile   131072
sonarqube   -   nproc    8192
EOF
vm.max_map_count=524288
fs.file-max=131072
vm.max_map_count = 524288
fs.file-max = 131072
```

Step 8. Creating dedicated user

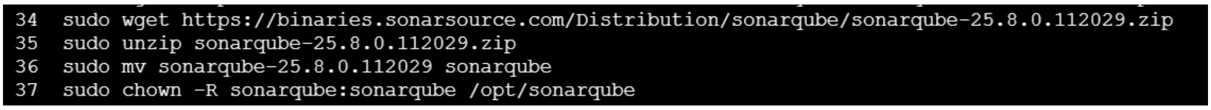


```
ubuntu@ip-172-31-33-208:~$ sudo useradd -r -s /bin/false sonarqube
```

Step 9. Download and Install SonarCube

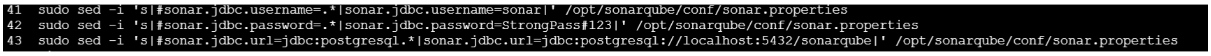


```
32  cd /opt
```



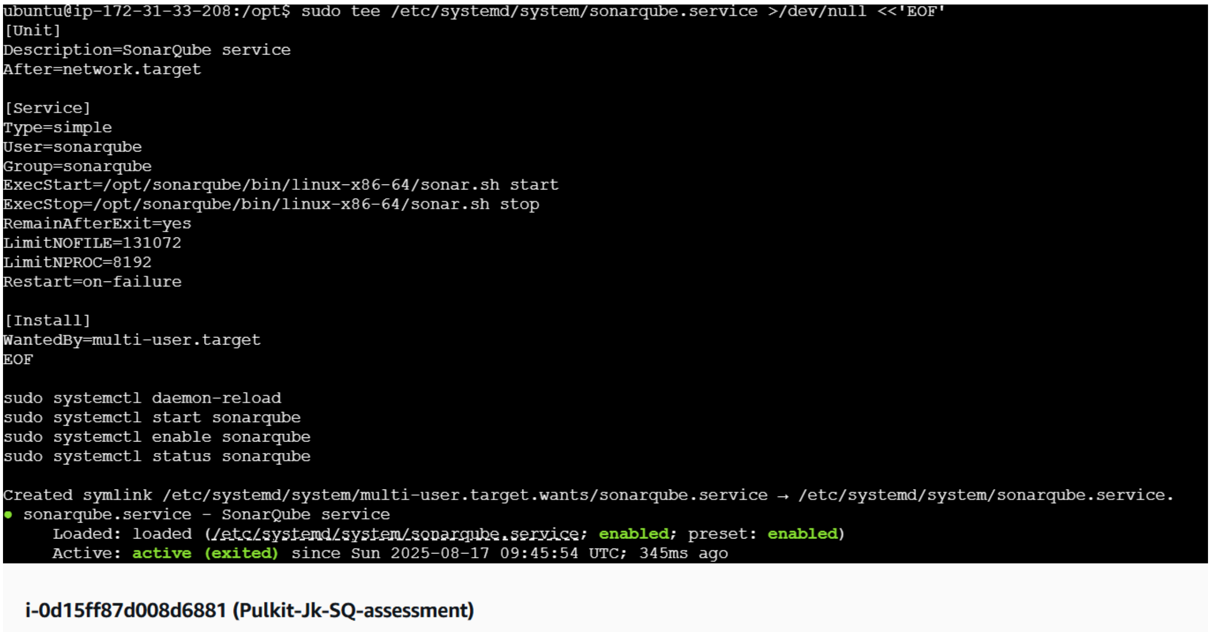
```
34  sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-25.8.0.112029.zip
35  sudo unzip sonarqube-25.8.0.112029.zip
36  sudo mv sonarqube-25.8.0.112029 sonarqube
37  sudo chown -R sonarqube:sonarqube /opt/sonarqube
```

Step 10. Configure DataBase in SonarCube



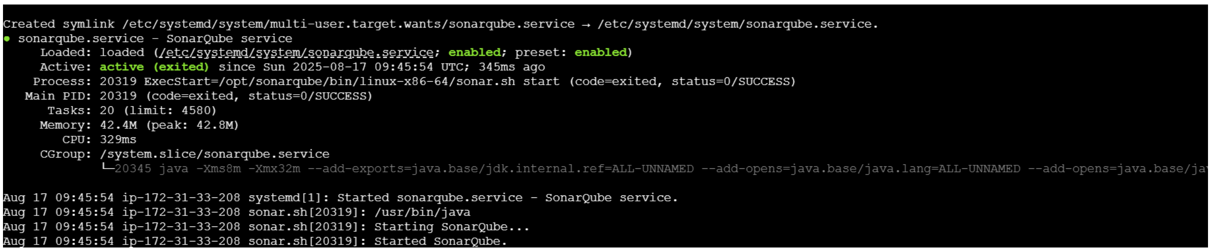
```
41  sudo sed -i 's|#sonar.jdbc.username=.*|sonar.jdbc.username=sonar|' /opt/sonarqube/conf/sonar.properties
42  sudo sed -i 's|#sonar.jdbc.password=.*|sonar.jdbc.password=StrongPass#123|' /opt/sonarqube/conf/sonar.properties
43  sudo sed -i 's|#sonar.jdbc.url=jdbc:postgresql.*|sonar.jdbc.url=jdbc:postgresql://localhost:5432/sonarqube|' /opt/sonarqube/conf/sonar.properties
```

Step 11. System Service



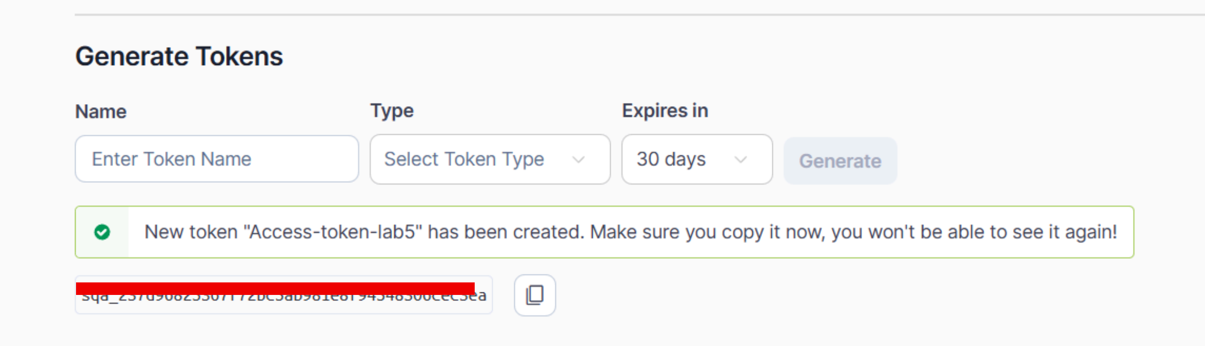
```
ubuntu@ip-172-31-33-208:/opt$ sudo tee /etc/systemd/system/sonarqube.service >/dev/null <<'EOF'
[Unit]
Description=SonarQube service
After=network.target

[Service]
Type=simple
User=sonarqube
Group=sonarqube
ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start
ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop
RemainAfterExit=yes
LimitNOFILE=131072
LimitNPROC=8192
Restart=on-failure

[Install]
WantedBy=multi-user.target
EOF

sudo systemctl daemon-reload
sudo systemctl start sonarqube
sudo systemctl enable sonarqube
sudo systemctl status sonarqube

Created symlink /etc/systemd/system/multi-user.target.wants/sonarqube.service → /etc/systemd/system/sonarqube.service.
● sonarqube.service - SonarQube service
     Loaded: loaded (/etc/systemd/system/sonarqube.service; enabled; preset: enabled)
     Active: active (exited) since Sun 2025-08-17 09:45:54 UTC; 345ms ago
```

i-0d15ff87d008d6881 (Pulkit-Jk-SQ-assessment)



```
Created symlink /etc/systemd/system/multi-user.target.wants/sonarqube.service → /etc/systemd/system/sonarqube.service.
● sonarqube.service - SonarQube service
     Loaded: loaded (/etc/systemd/system/sonarqube.service; enabled; preset: enabled)
     Active: active (exited) since Sun 2025-08-17 09:45:54 UTC; 345ms ago
    Process: 20319 ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start (code=exited, status=0/SUCCESS)
   Main PID: 20319 (code=exited, status=0/SUCCESS)
      Tasks: 20 (limit: 4580)
     Memory: 42.4M (peak: 42.8M)
        CPU: 329ms
     CGroup: /system.slice/sonarqube.service
             └─20345 java -Xms8m -Xmx32m --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/ja

Aug 17 09:45:54 ip-172-31-33-208 systemd[1]: Started sonarqube.service - SonarQube service.
Aug 17 09:45:54 ip-172-31-33-208 sonar.sh[20319]: /usr/bin/java
Aug 17 09:45:54 ip-172-31-33-208 sonar.sh[20319]: Starting SonarQube...
Aug 17 09:45:54 ip-172-31-33-208 sonar.sh[20319]: Started SonarQube.
```

Step 12. Generated access token



Generate Tokens

Name: Enter Token Name

Type: Select Token Type

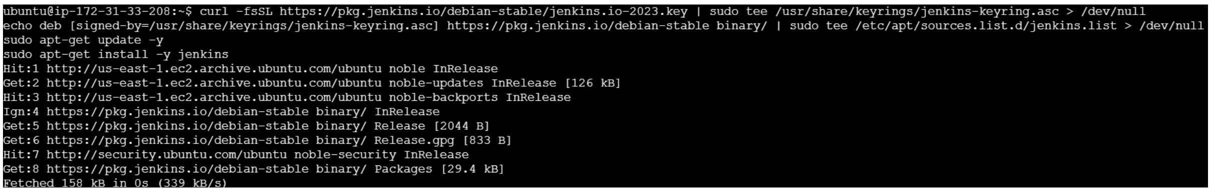
Expires in: 30 days

New token "Access-token-lab5" has been created. Make sure you copy it now, you won't be able to see it again!

[redacted]

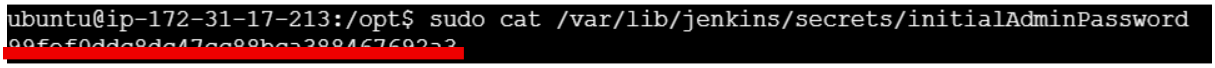
Part B

Step 1. Install Jenkins



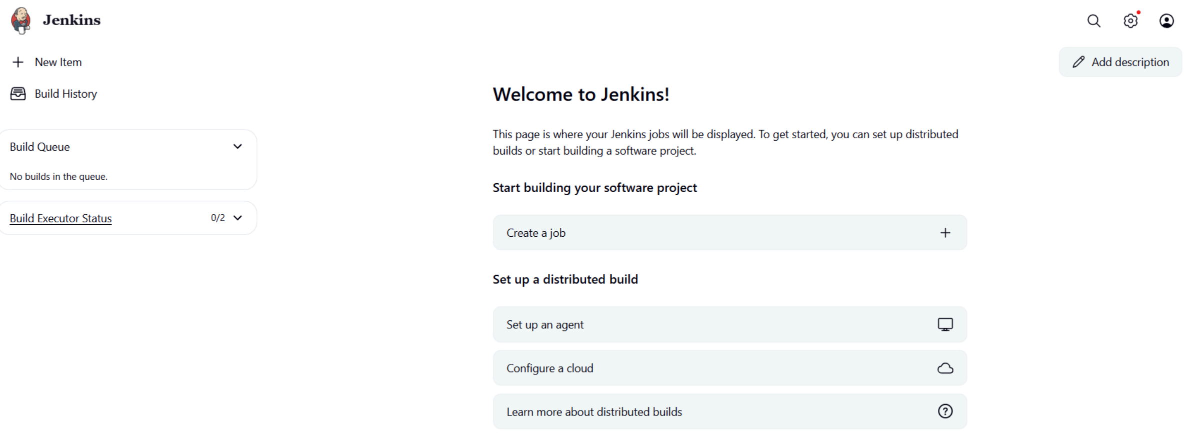
```
ubuntu@ip-172-31-33-208:~$ curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update -y
sudo apt-get install -y jenkins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:7 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [29.4 kB]
Fetched 158 kB in 0s (339 kB/s)
```

Step 2. Getting Jenkins Password to login



```
ubuntu@ip-172-31-17-213:/opt$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
[redacted]
```

Step 3. Logging in Jenkins



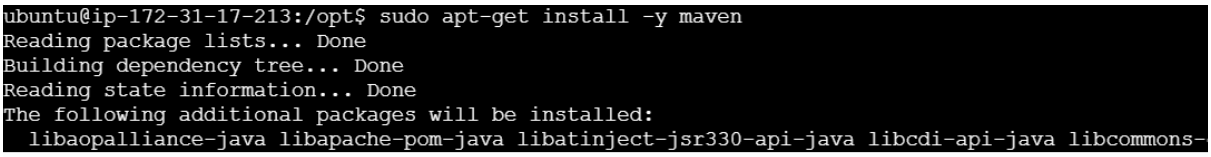
Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

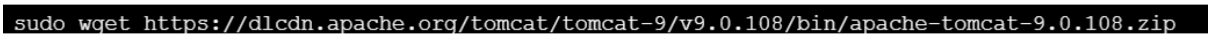
Set up a distributed build

Step 4. Installing Maven

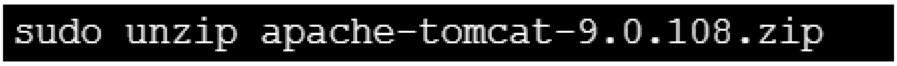


```
ubuntu@ip-172-31-17-213:/opt$ sudo apt-get install -y maven
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libaopalliance-java libapache-pom-java libatinject-jsr330-api-java libcdi-api-java libcommons-
```

Step 5. Installing Apache tomcat



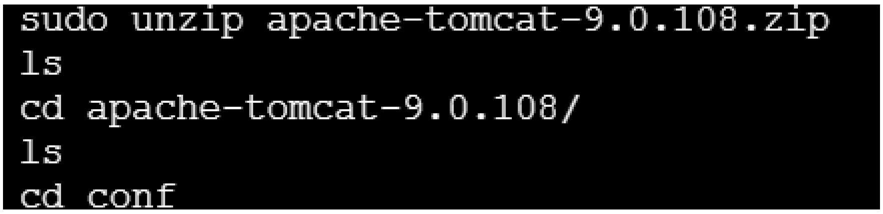
```
 sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.108/bin/apache-tomcat-9.0.108.zip
```



```
sudo unzip apache-tomcat-9.0.108.zip
```

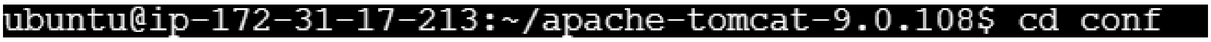
Step 6. Since Tomcat and Jenkins both run at 8080 port there will be clash thus we need to change working port of tomcat

a) Going into conf directory to work in server.xml file (cd apache-tomcat-9.0.108./conf)

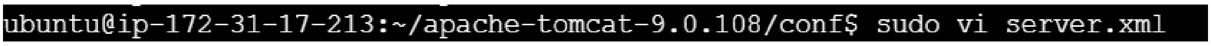


```
sudo unzip apache-tomcat-9.0.108.zip
ls
cd apache-tomcat-9.0.108/
ls
cd conf
```

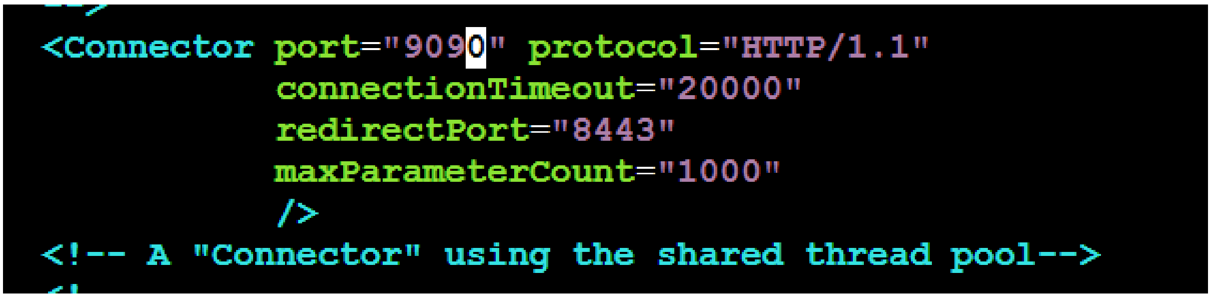
b) changing port to 9090



```
ubuntu@ip-172-31-17-213:~/apache-tomcat-9.0.108$ cd conf
```

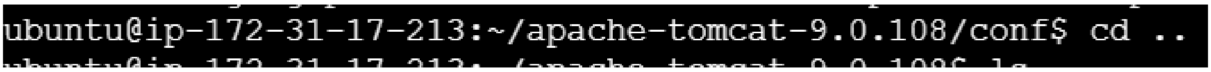


```
ubuntu@ip-172-31-17-213:~/apache-tomcat-9.0.108/conf$ sudo vi server.xml
```

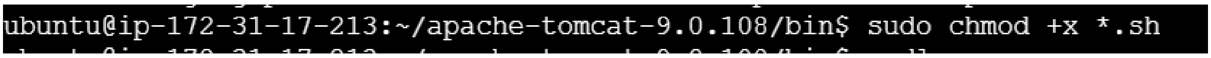


```
  <Connector port="9090" protocol="HTTP/1.1"
             connectionTimeout="20000"
             redirectPort="8443"
             maxParameterCount="1000"
             />
  <!-- A "Connector" using the shared thread pool-->
```

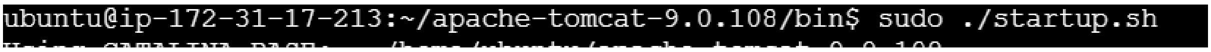
c)Moving to bin directory to start startup.sh file



```
ubuntu@ip-172-31-17-213:~/apache-tomcat-9.0.108/conf$ cd ..
```

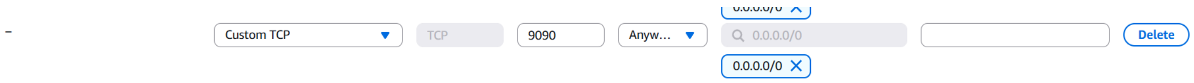


```
ubuntu@ip-172-31-17-213:~/apache-tomcat-9.0.108/bin$ sudo chmod +x *.sh
```



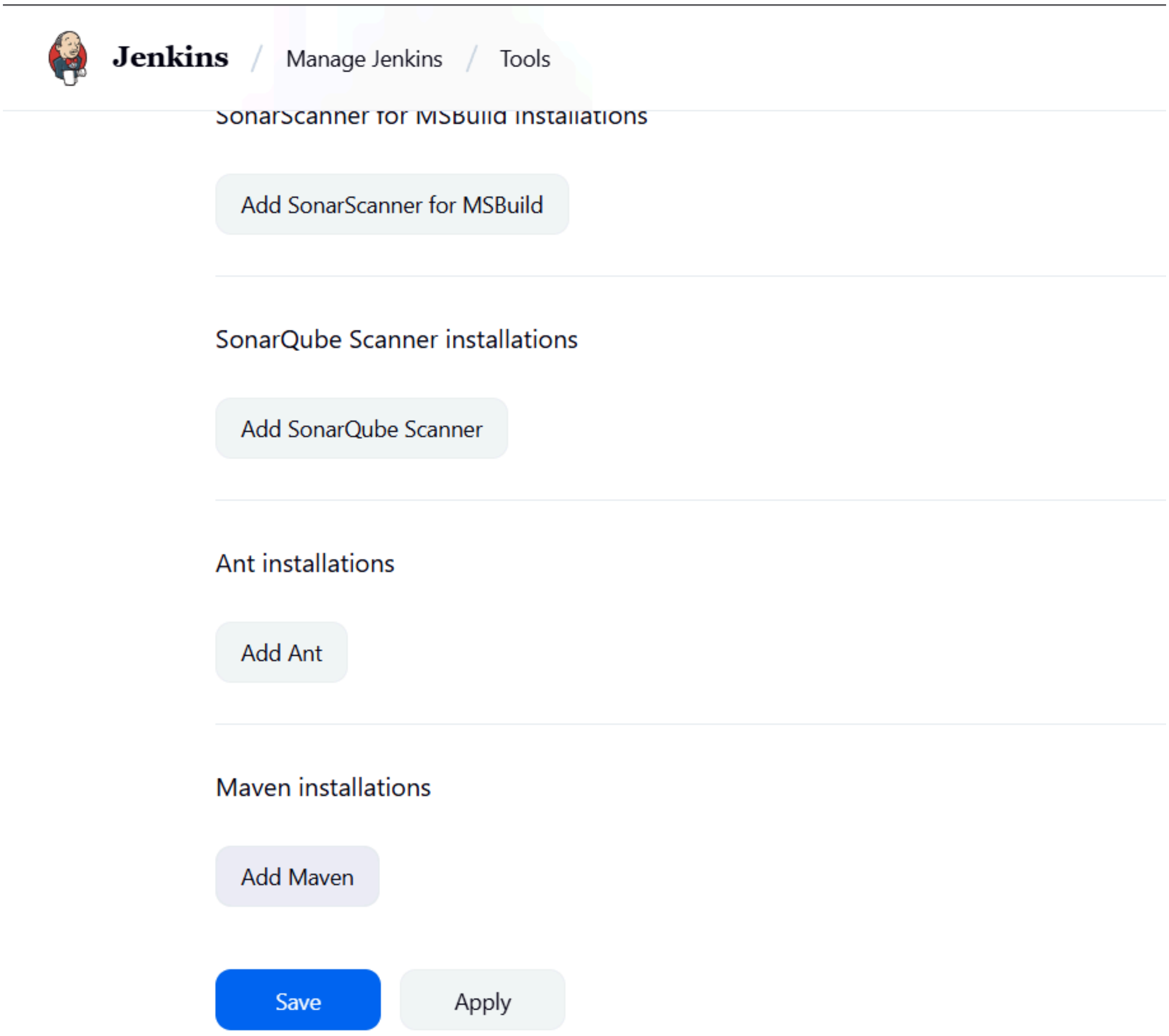
```
ubuntu@ip-172-31-17-213:~/apache-tomcat-9.0.108/bin$ sudo ./startup.sh
```

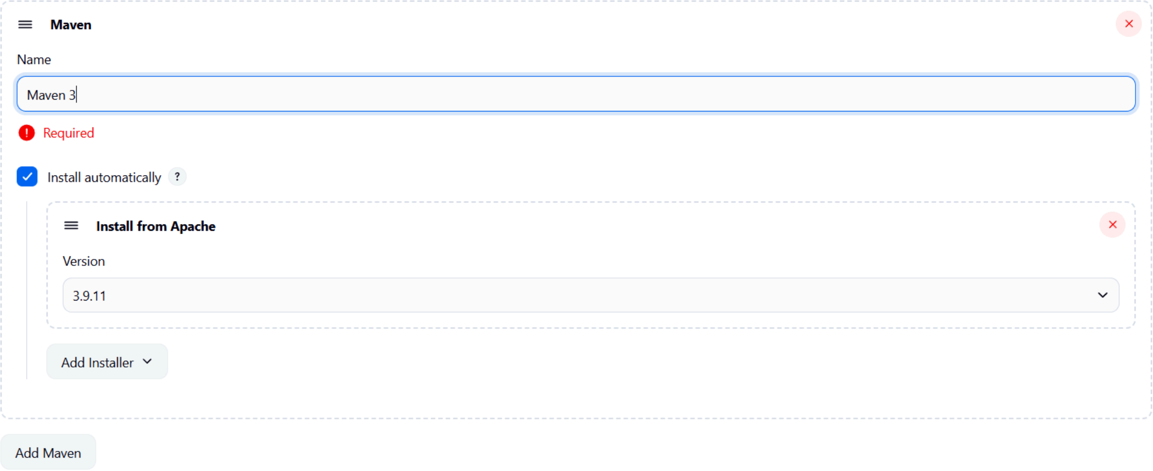
d) Since we have changes the port to 9090 thus we need to edit inbound rule in security group



Custom TCP	TCP	9090	Anyw...	0.0.0.0/0 0.0.0.0/0	

Step 4. Attaching maven to Jenkins (Jenkins-> manage Jenkins -> tools ->maven installations)


Jenkins / Manage Jenkins / Tools
SonarScanner for MSBuild installations
Add SonarScanner for MSBuild
SonarQube Scanner installations
Add SonarQube Scanner
Ant installations
Add Ant
Maven installations
Add Maven
Save
Apply



Name: Maven 3

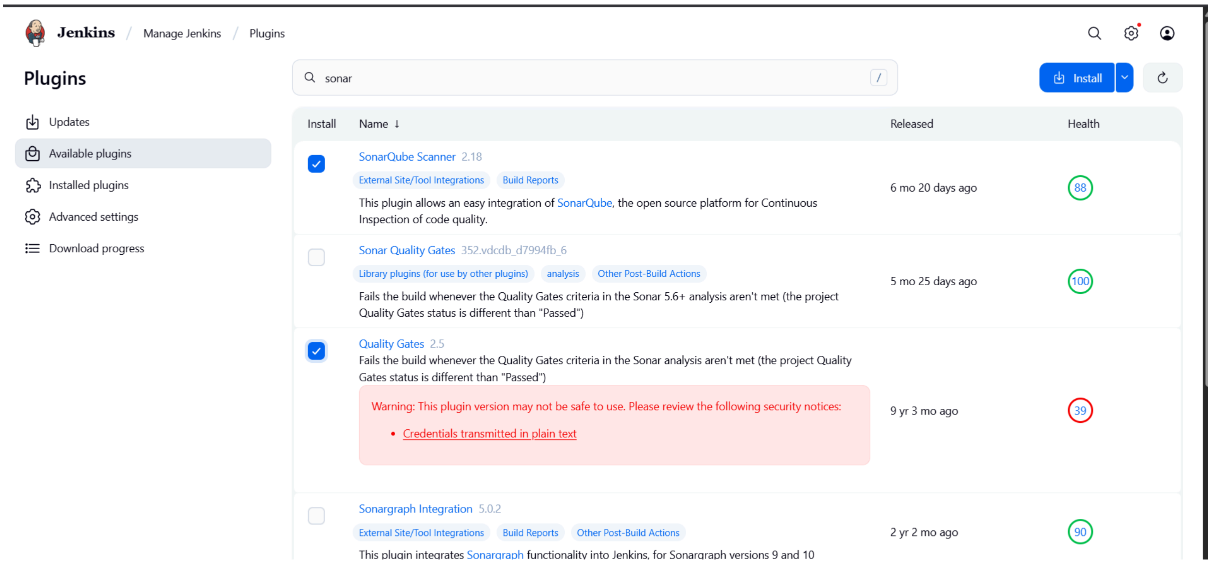
Required

☑ Install automatically ?

Install from Apache

Version: 3.9.11

Step 5. Installing plugins



Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

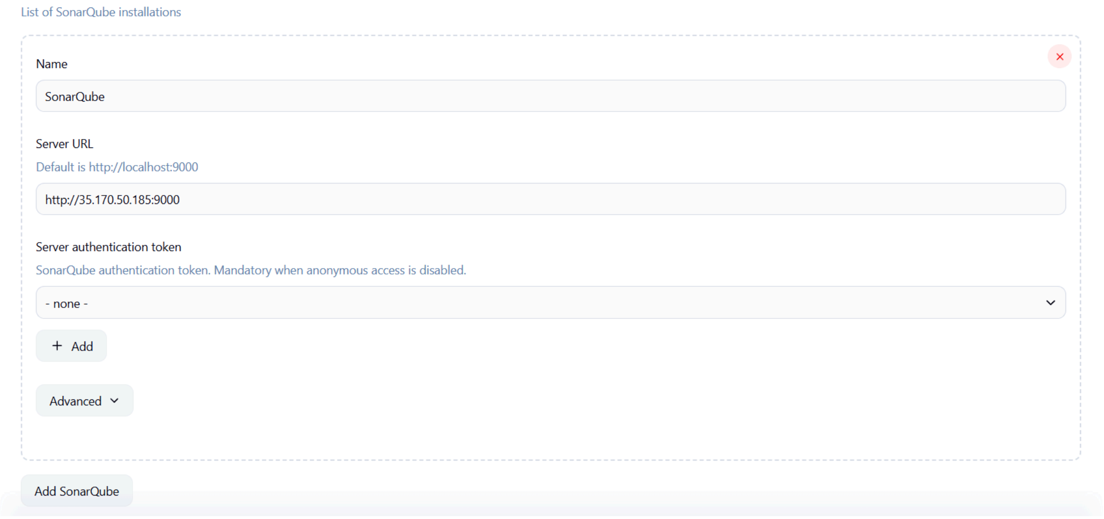
sonar

Install

Install	Name ↓	Released	Health
☑	SonarQube Scanner 2.18 External Site/Tool Integrations Build Reports This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	6 mo 20 days ago	88
□	Sonar Quality Gates 352.vdcdb_d7994fb_6 Library plugins (for use by other plugins) analysis Other Post-Build Actions Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")	5 mo 25 days ago	100
☑	Quality Gates 2.5 Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed") Warning: This plugin version may not be safe to use. Please review the following security notices: • Credentials transmitted in plain text	9 yr 3 mo ago	39
□	Sonargraph Integration 5.0.2 External Site/Tool Integrations Build Reports Other Post-Build Actions This plugin integrates Sonargraph functionality into Jenkins, for Sonargraph versions 9 and 10	2 yr 2 mo ago	90

Step 6. Adding SonarQube Server into Jenkins

a) Adding Sonar Server


List of SonarQube installations

Name

SonarQube

Server URL

Default is http://localhost:9000

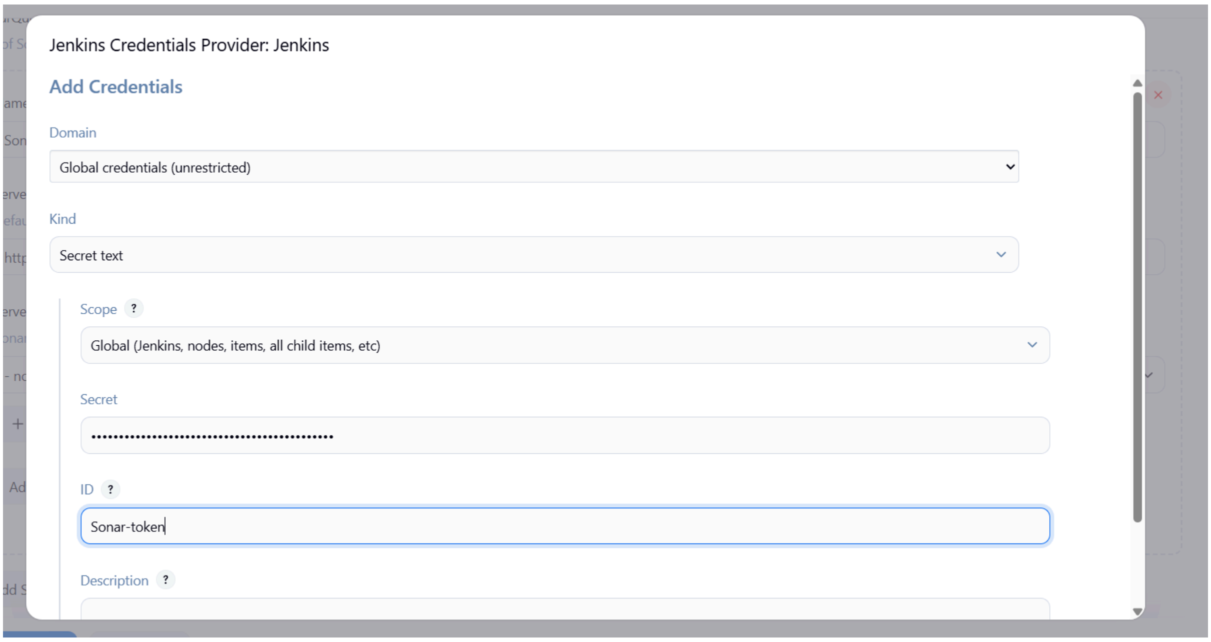
http://35.170.50.185:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

b) Creating a Secret text (Jenkins Credentials Provider)



Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Secret text

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Secret

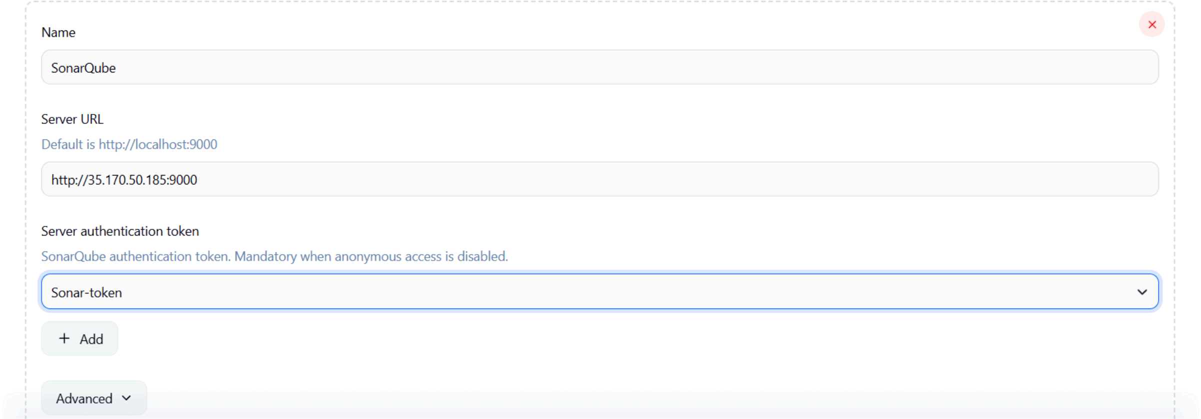
••

ID ?

Sonar-token

Description ?

c) Saving the Sonar Token and then the Server



Name

SonarQube

Server URL

Default is http://localhost:9000

http://35.170.50.185:9000

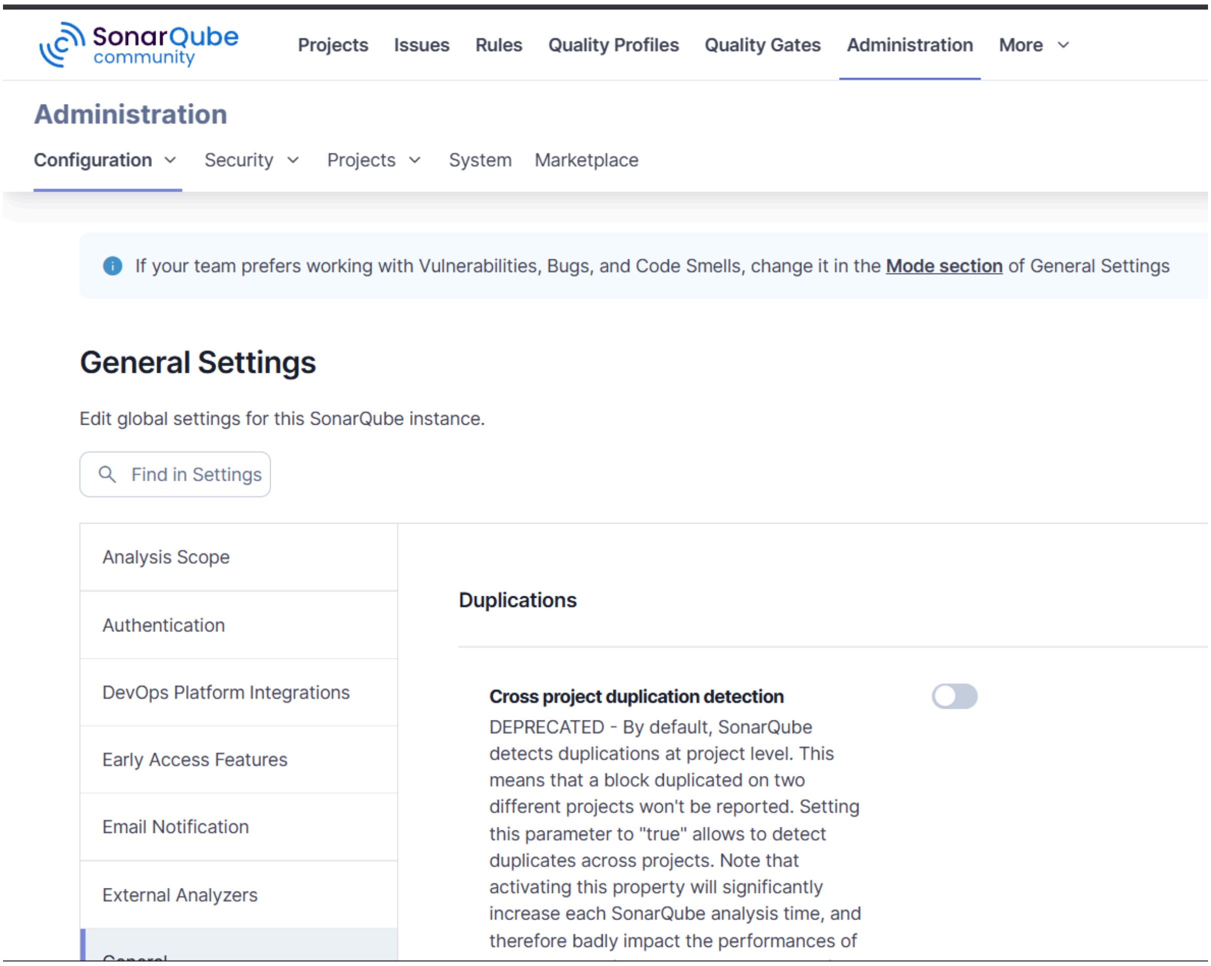
Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

Sonar-token

Step 7. Adding Jenkins webhook on SonarQube

a) Administration -> Configuration -> Webhook -> Create



If your team prefers working with Vulnerabilities, Bugs, and Code Smells, change it in the **Mode section** of General Settings

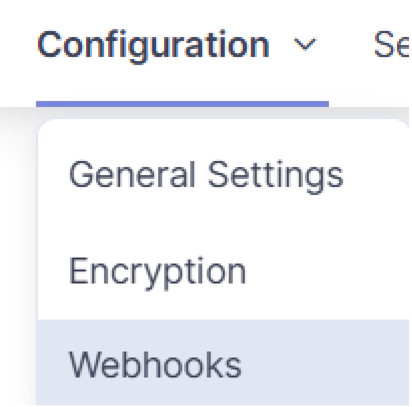
General Settings

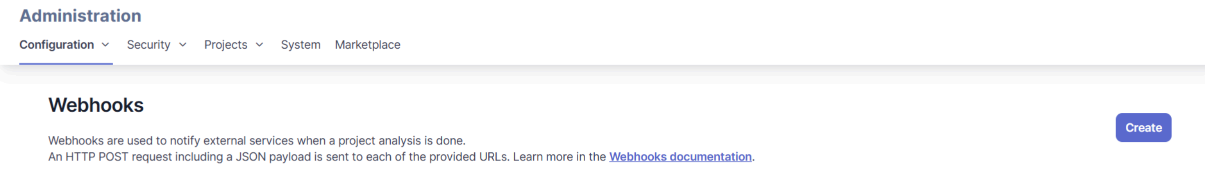
Edit global settings for this SonarQube instance.

Duplications

Cross project duplication detection

DEPRECATED - By default, SonarQube detects duplications at project level. This means that a block duplicated on two different projects won't be reported. Setting this parameter to "true" allows to detect duplicates across projects. Note that activating this property will significantly increase each SonarQube analysis time, and therefore badly impact the performances of

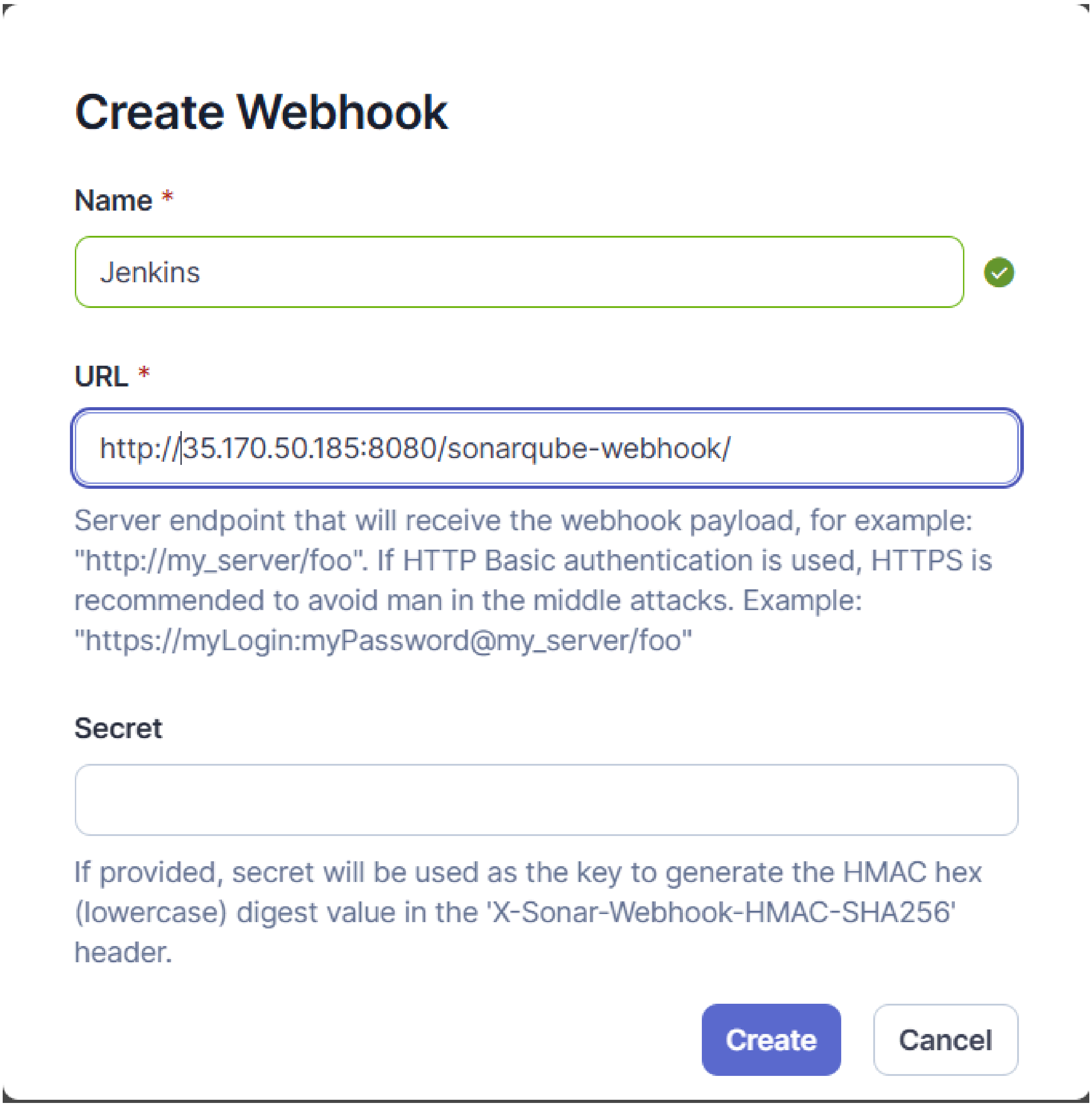




Webhooks

Webhooks are used to notify external services when a project analysis is done.
An HTTP POST request including a JSON payload is sent to each of the provided URLs. Learn more in the Webhooks documentation.

b) Create webhook



Create Webhook

Name *

Jenkins

URL *

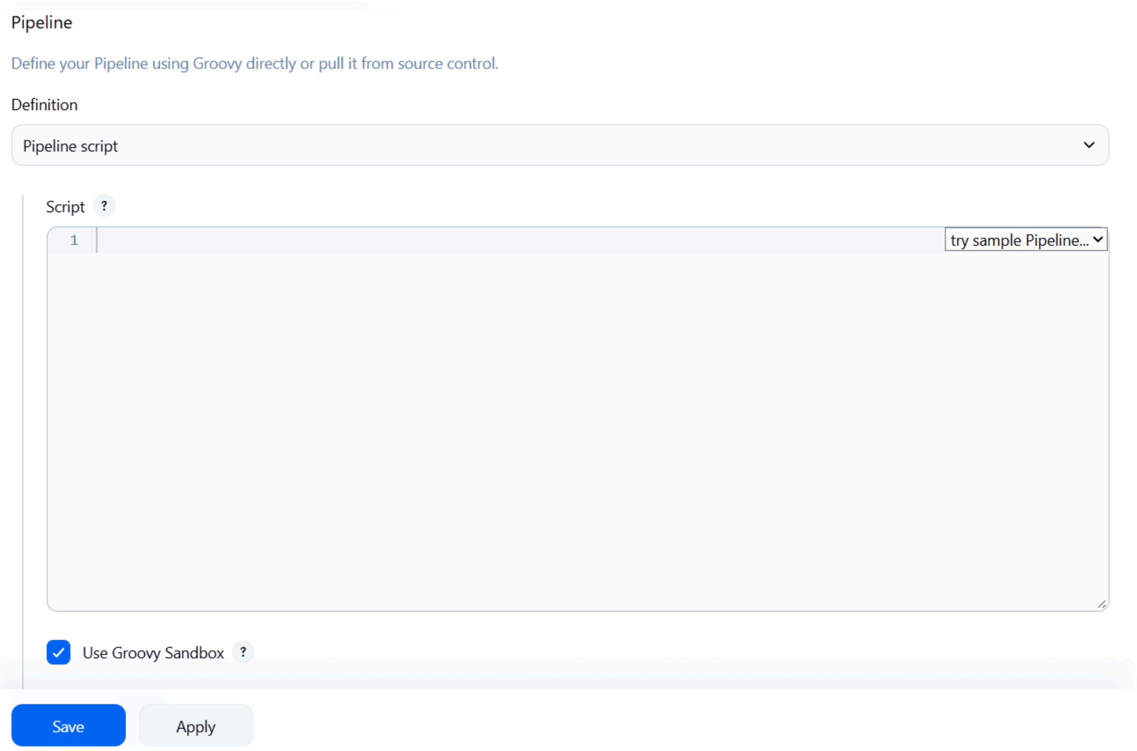
http://35.170.50.185:8080/sonarqube-webhook/

Server endpoint that will receive the webhook payload, for example: "http://my_server/foo". If HTTP Basic authentication is used, HTTPS is recommended to avoid man in the middle attacks. Example: "https://myLogin:myPassword@my_server/foo"

Secret

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header.

Step 8. Creating a Pipeline in Jenkins



Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

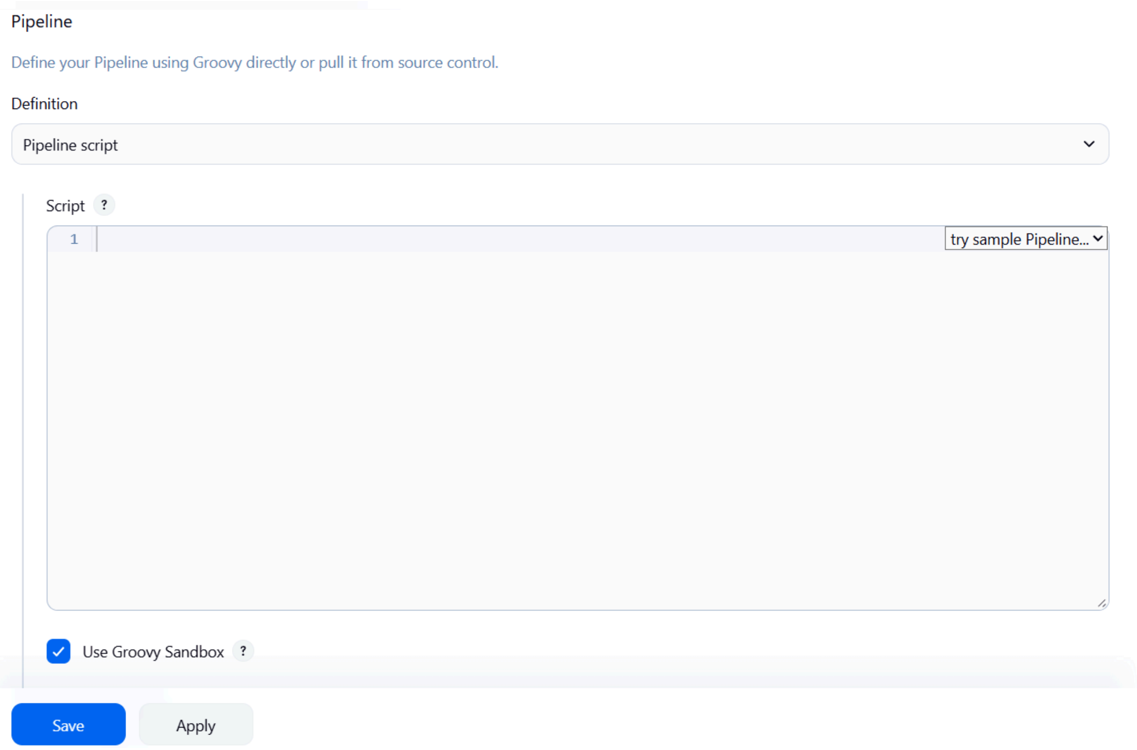
Definition

Pipeline script

Script

☑ Use Groovy Sandbox

Step 9. Writing the groovy script



Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

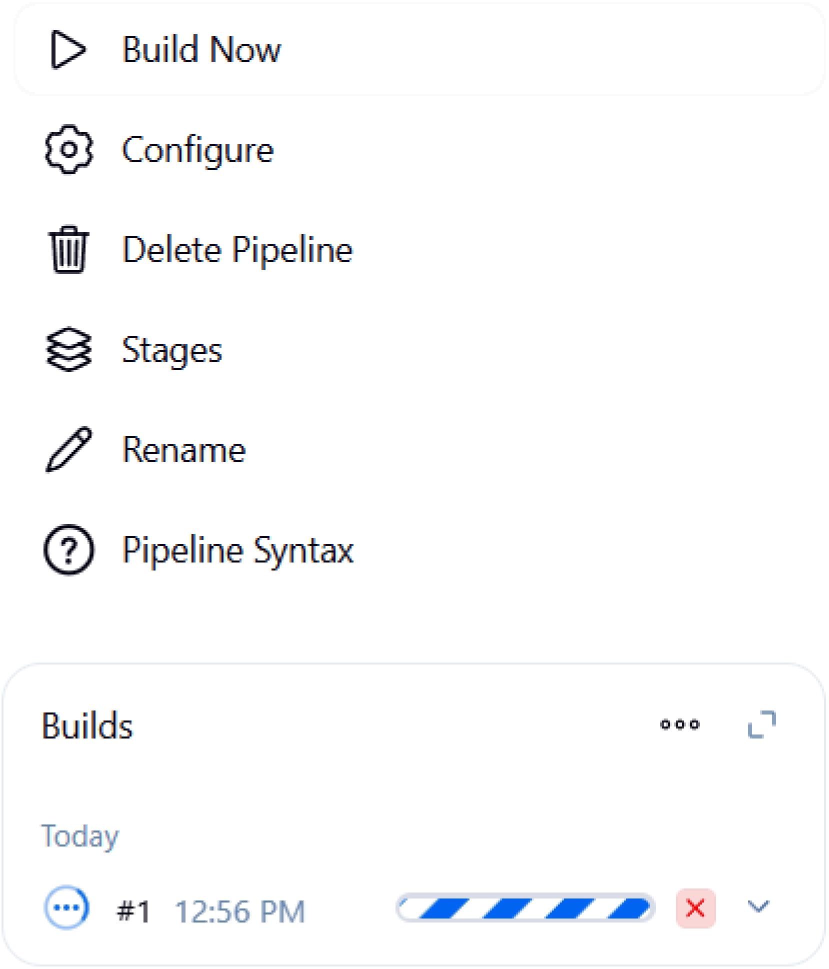
Pipeline script

Script

☑ Use Groovy Sandbox

Step 10. Build now

Per

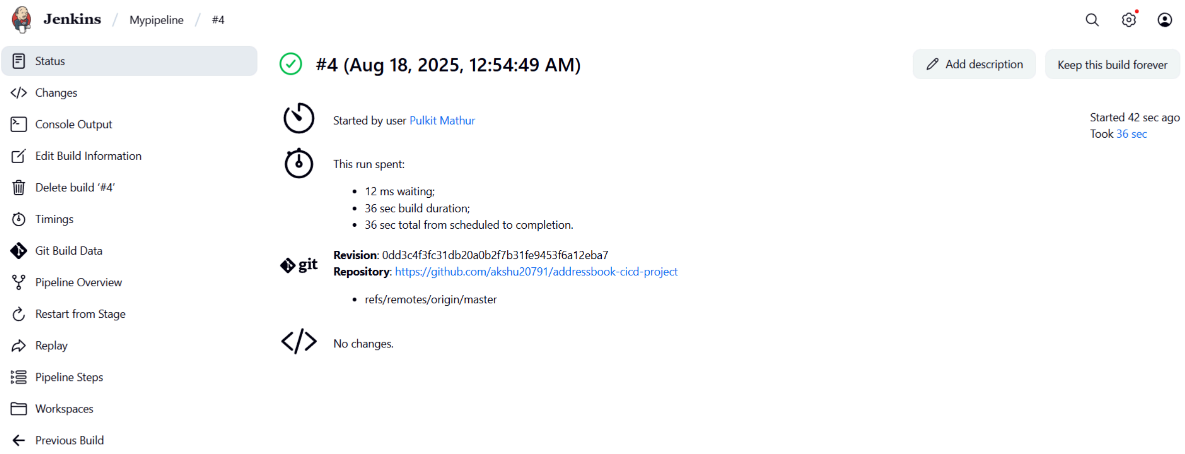


Builds

Today

#1 12:56 PM

Step 11 Tackling errors and successfully building pipeline



#4 (Aug 18, 2025, 12:54:49 AM)

Started by user Pulkit Mathur

Started 42 sec ago
Took 36 sec

This run spent:

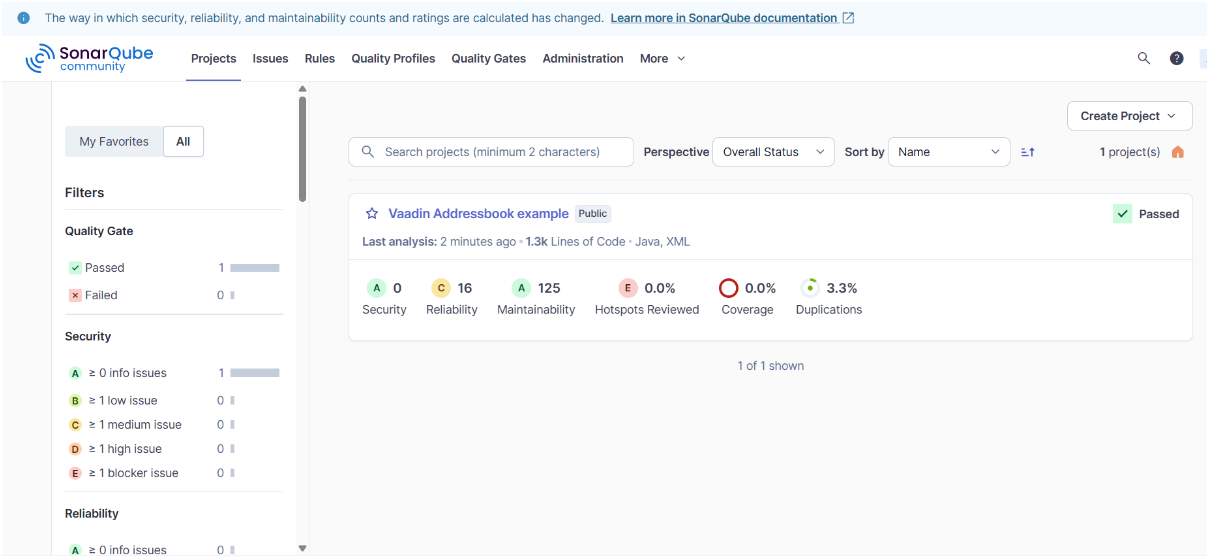
- 12 ms waiting;
- 36 sec build duration;
- 36 sec total from scheduled to completion.

Revision: 0dd3c4f3fc31db20a0b2f7b31fe9453f6a12eba7
Repository: https://github.com/akshu20791/addressbook-cicd-project

- refs/remotes/origin/master

No changes.

Step 12. Review QA report in SonarQube



The way in which security, reliability, and maintainability counts and ratings are calculated has changed. Learn more in SonarQube documentation

Perspective Overall Status Sort by Name 1 project(s)

Filters

Quality Gate

✓ Passed	1
✗ Failed	0

Security

A ≥ 0 info issues	1
B ≥ 1 low issue	0
C ≥ 1 medium issue	0
D ≥ 1 high issue	0
E ≥ 1 blocker issue	0

Reliability

A ≥ 0 info issues	0

Vaadin Addressbook example Public ✓ Passed

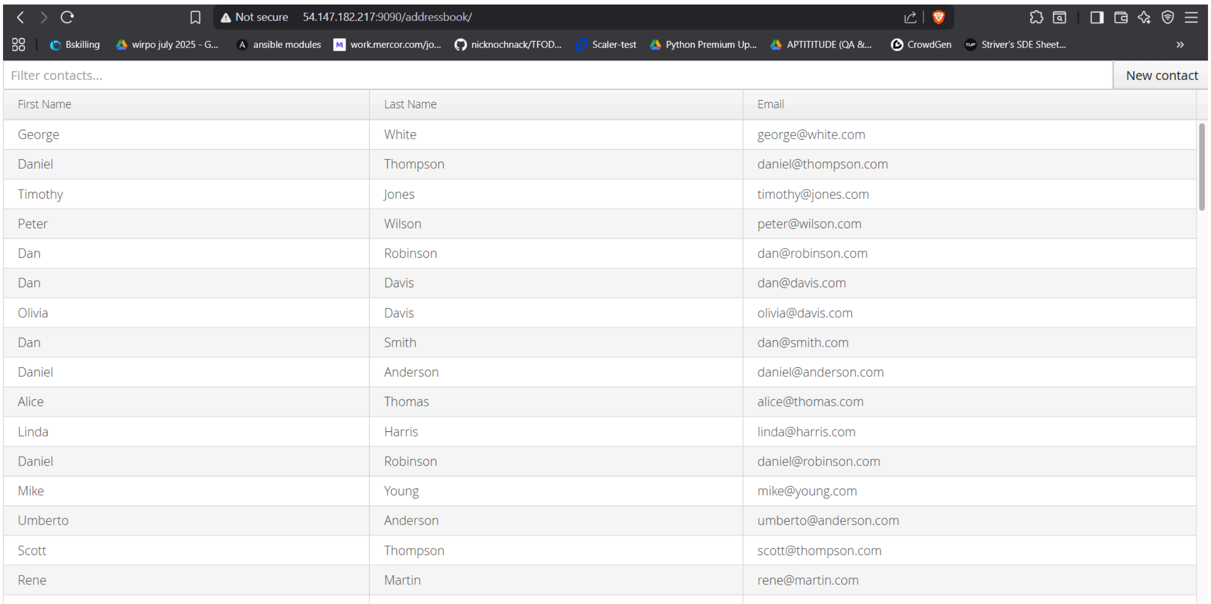
Last analysis: 2 minutes ago • 1.3k Lines of Code • Java, XML

A 0	C 16	A 125	E 0.0%	0.0%	3.3%
Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications

1 of 1 shown

Step 13 Verification of addressbook deployment

http://<public_ip>:9090/addressbook/



First Name	Last Name	Email
George	White	george@white.com
Daniel	Thompson	daniel@thompson.com
Timothy	Jones	timothy@jones.com
Peter	Wilson	peter@wilson.com
Dan	Robinson	dan@robinson.com
Dan	Davis	dan@davis.com
Olivia	Davis	olivia@davis.com
Dan	Smith	dan@smith.com
Daniel	Anderson	daniel@anderson.com
Alice	Thomas	alice@thomas.com
Linda	Harris	linda@harris.com
Daniel	Robinson	daniel@robinson.com
Mike	Young	mike@young.com
Umberto	Anderson	umberto@anderson.com
Scott	Thompson	scott@thompson.com
Rene	Martin	rene@martin.com